Applied Mathematical Modelling 37 (2013) 570–585

Contents lists available at SciVerse ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

A novel algorithm based on parameterization method for calculation of curvature of the free surface flows

H. Saghi [a], M.J. Ketabdari [b,*], M. Zamirian [c]

[a] Department of Civil Engineering, Ferdowsi University, Mashhad, Iran
[b] Faculty of Marine Technology, Amirkabir University of Technology, Tehran, Iran
[c] Department of Mathematics, Azad University of Bojnourd, Bojnourd, Iran

ARTICLE INFO

Article history:
Received 22 May 2011
Received in revised form 17 February 2012
Accepted 29 February 2012
Available online 9 March 2012

Keywords:
Parameterization method
Curvature
Free surface flows
Surface tension
Volume of fluid

ABSTRACT

In this paper, a new approach based on parameterization method is presented for calculation of curvature on the free surface flows. In some phenomena such as droplet and bubble, surface tension is prominent. Therefore in these cases, accurate estimation of the curvature is vital. Volume of fluid (VOF) is a surface capturing method for free surface modeling. In this method, free surface curvature is calculated based on gradient of scalar transport parameter which is regarded as original method in this paper. However, calculation of curvature for a circle and other known geometries based on this method is not accurate. For instance, in practice curvature of a circle in interface cells is constant, while this method predicts different curvatures for it. In this research a novel algorithm based on parameterization method for improvement of the curvature calculation is presented. To show the application of parameterization method, two methods are employed. In the first approach denoted by, three line method, a curve is fitted to the free surface so that the distance between curve and linear interface approximation is minimized. In the second approach namely four point method, a curve is fitted to intersect points with grid lines for central and two neighboring cells. These approaches are treated as calculus of variation problems. Then, using the parameterization method, these cases are converted into the sequences of time-varying nonlinear programming problems. With some treatments a conventional equivalent model is obtained. It is finally proved that the solution of these sequences in the models tends to the solution of the calculus of variation problems. For verification of the presented methods, curvature of some geometrical shapes such as circle, elliptic and sinusoidal profile is calculated and compared with original method used in VOF process and analytical solutions. Finally, as a more practical problem, spurious currents are studied. The results showed that more accurate curve prediction is obtained by these approaches than the original method in VOF approach.

1. Introduction

Motion of fluids with a free surface such as droplet splashing and bubble, are important phenomena in many fields of fluid mechanics. Therefore, some researches have been focused on solving this problem using different numerical techniques. Since the location of the free surface is driven by the gross motions of the fluid, accurate simulation of such phenomena is very cumbersome. To simulate this problem, precise modeling of surface tension is vital. In these problems, interface

* Corresponding author. Address: 424 Hafez Avenue, P.O. Box 15875-4413, Tehran, Iran.
E-mail address: Ketabdar@aut.ac.ir (M.J. Ketabdari).

0307-904X/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.apm.2012.02.043

normal vectors and curvatures, are required to model surface tension. Volume of fluid method is one the surface capturing techniques for interface modeling of two phase flows. There are different methods for modeling of curvature. The most widely used technique is calculation of the spatial derivatives of the scalar function based on VOF at any instant. In this method, the gradient of the scalar function [1] is normal to the interface. Then, by taking the divergence of this interface vector, the second derivative of the scalar function, i.e. the interface curvature is obtained. The methods of discretization of the surface tension are associated with the interface curvature. They are the main source of error in surface tension simulation [2]. To reduce this error, various researches have been performed [2–5]. For example, Francois et al. [2] impose an exact balance between the surface tension and pressure forces to model surface tension into a volume-of-fluid (VOF) method and so, no spurious currents are induced in a flow provided. Some researchers employed the VOF-based height-function method [2,6,7] to obtain the curvatures with second-order accuracy. In this method, the interface curvature is obtained from the derivatives of the height function. However, it leads to poor results, if an interface is not adequately resolved [8]. Poo and Ashgriz [9] utilized a second-order polynomial to calculate the curvature. In a method known as PROST [10], the data is fitted iteratively with a 2D or 3D parabola.

Surface tension along with an interface arises from forces between molecules in a fluid. Surface tension is important when Webber number (We) is much greater than unity ($We \gg 1$) in high Reynolds numbers ($\mathrm{Re} \gg 1$) or when Capillary number (Ca) is much lower that unity ($Ca \ll 1$) for small Reynolds numbers ($\mathrm{Re} \ll 1$). These parameters are defined as:

$$We = \frac{\rho L U^2}{\sigma}, \tag{1}$$

$$Ca = \frac{\mu U}{\sigma}, \tag{2}$$

$$\mathrm{Re} = \frac{\rho U L}{\mu}, \tag{3}$$

where σ is the surface tension, ρ is the fluid density, U is the velocity scale, L is the length scale and μ is the fluid dynamic viscosity. The focus of this paper is calculation of curvature based on the parameterization method (PM). To illustrate this new approach, the volume-of-fluid method is used to represent the interface. Then intersection points of interface and grid lines are determined. Calculus of variation problems (CVP) is achieved using three line method (TLM) and four point method (FPM) approaches. The solutions of these CVPs are the Optimal Curve (OC) in the form of $f(\cdot)$. In fact $f(\cdot)$ is the approximation of surface flow. Substituting the sequence of polynomials, $p_n(\cdot)$, $n = 1, 2, \ldots$, instead of $f(\cdot)$ in the CVPs (PM), the sequence of Time-varying Nonlinear Programming Problems (TNLPP) is achieved. It should be noted that variables of TNLPPs are the constant coefficients of polynomials. Sequence of TNLPPs can be converted to the sequence of Nonlinear Programming Problems (NLPP) with some calculations. It is proved that the NLPPs solution tends to the solution of CVP. Finally, NLPPs solution leads to two polynomials as the OCs of the FPM and TLM. The accuracy and performance of the new method are demonstrated via numerical test cases with known curvatures.

2. Curvature simulation

In this section, the interface cell curvature was modeled with original (VOF) and PM methods.

2.1. Original (VOF) method

The successful approaches for handling free surface problems can be categorized as surface tracking and surface capturing methods. Surface tracking methods try to solve the flow in the fluid region while the free surface is treated as a moving boundary of the computational domain. This usually satisfies the kinematic boundary conditions. In this technique, free surface locations can be determined precisely. Surface capturing methods simulate both fluid regions on a fixed grid system. In these methods, the free surface can be identified using a marker function such as the marker particles in the marker and cell (MAC) method [11] or the volume fraction in VOF method [12–14]. To determine the volume fraction of each phase (e.g. air and water) in VOF method, a scalar transport equation which is colour function (F) is solved in all computational cells as [15]:

$$\frac{\partial F}{\partial t} + \vec{\nabla} \cdot (F\vec{U}) = 0. \tag{4}$$

So that:

$$\begin{cases} F = 1 & \textit{for cells inside fluid } 1, \\ F = 0 & \textit{for cells inside fluid } 2, \\ 0 < F < 1 & \textit{for free surface cells}. \end{cases} \tag{5}$$

The interface unit normal vector and curvature of free surface can be calculated from the gradient of F as:

$$\hat{n} = \frac{\nabla F}{|\nabla F|}, \tag{6}$$

$$\kappa = -\nabla \cdot \hat{n} = -\nabla \cdot \left(\frac{\nabla F}{|\nabla F|}\right), \tag{7}$$

where κ is the interface curvature. A schematic interface is shown in Fig. 1(a). It is the boundary between two fluids. The volume fractions of cells are presented in Fig. 1(b).

In this paper, Youngs' VOF (Y-VOF) method is used as based method to model free surface and curvature estimation. In this method, the interface is approximated by a straight line segment with orientation β which cut the cell. Four possible interface reconstructions exist in this method as presented in Fig. 2:

To calculate β, normal vectors on surface are estimated as:

$$\beta_{i,j} = \tan^{-1}\left[\frac{-n^x_{i,j}}{n^y_{i,j}}\right], \quad -\pi < \beta_{i,j} < \pi, \tag{8}$$

$$n^x_{i,j} = \frac{1}{\Delta x}(F_{i+1,i+1} + 2F_{i+1,i} + F_{i+1,i-1} - F_{i-1,i+1} - 2F_{i-1,i} - F_{i-1,i-1}), \tag{9}$$

$$n^y_{i,j} = \frac{1}{\Delta x}(F_{i+1,i+1} + 2F_{i,i+1} + F_{i+1,i+1} - F_{i+1,i-1} - 2F_{i,i-1} - F_{i-1,i-1}), \tag{10}$$

where Δx and Δy are the mesh sizes in x and y directions and $F_{i,j}$ is the colour function. In some problems such as bubble rises and droplet splashing, the volume fractions vary sharply from zero to one across the interface. This discontinuous behavior decreases the accuracy of estimation of the first and second derivatives of F. This leads to inaccurate interface normal vector and therefore approximate curvature calculations. It should be noted that, the results can be improved if ∇F is smoothed [8]. The expression for cell-center curvature evaluation can be given as [16]:

$$\kappa = \frac{\left(\frac{n_{x,i+\frac{1}{2},j+\frac{1}{2}}+n_{x,i+\frac{1}{2},j-\frac{1}{2}}-n_{x,i-\frac{1}{2},j+\frac{1}{2}}-n_{x,i-\frac{1}{2},j-\frac{1}{2}}}{2}\right)}{\Delta x} + \frac{\left(\frac{n_{y,i+\frac{1}{2},j+\frac{1}{2}}+n_{y,i-\frac{1}{2},j+\frac{1}{2}}-n_{y,i+\frac{1}{2},j-\frac{1}{2}}-n_{y,i-\frac{1}{2},j-\frac{1}{2}}}{2}\right)}{\Delta y}, \tag{11}$$

where Δx and Δy are the mesh sizes in x and y directions respectively. $n_{x,i+\frac{1}{2},j+\frac{1}{2}}$ and $n_{y,i+\frac{1}{2},j+\frac{1}{2}}$ are the components of normal unit vector in $i+\frac{1}{2}, j+\frac{1}{2}$ as:

$$n_{x,i+\frac{1}{2},j+\frac{1}{2}} = \frac{\left(\frac{F_{i+1,j+1}+F_{i+1,j}-F_{i,j+1}-F_{i,j}}{2}\right)}{\Delta x}, \tag{12}$$

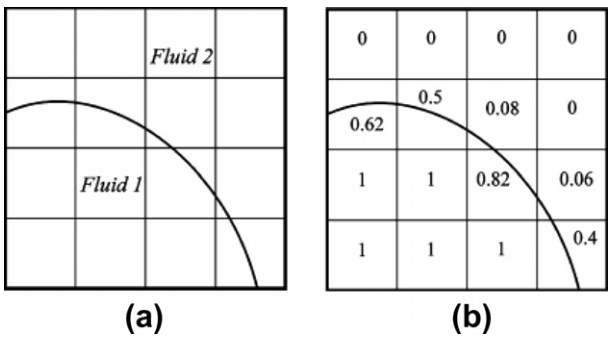

Fig. 1. Interface representation: (a) Schematic interface and (b) volume fraction distribution in VOF method.

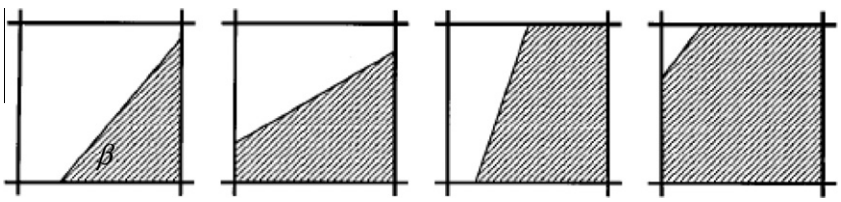

Fig. 2. Four possible interface reconstruction for Y-VOF method.

$$n_{y,i+\frac{1}{2},j+\frac{1}{2}} = \frac{\left(\frac{F_{i+1,j+1}+F_{i,j+1}-F_{i+1,j}-F_{i,j}}{2}\right)}{\Delta y}. \tag{13}$$

Similar equations can be written for other parameters.

2.2. PM method

The PM method is associated with two approaches denoted by FPM and TLM.

2.3. FPM method

In this method, a function as $f(\cdot) \in C^2(x_1,x_4)$, (the set of functions with continuous second derivation) is found to minimize the distance between $f(\cdot)$ and points y_i $(i = 1,2,3,4)$ such that $\kappa(x) = \frac{|\ddot{f}(x)|}{\sqrt{1+\dot{f}^2(x)}}$ is an approximately constant number in every interval $(x_i,x_{i+1}]$. To model this problem, the following CVP is achieved:

$$\inf_{f(\cdot)\in C^2(x_1,x_4)} I(f(\cdot)) = \sum_{i=1}^{4} |f(x_i) - y_i| \tag{14}$$

$$S.t. \begin{cases} |\kappa(x) - C_1| \leqslant \varepsilon, & x \in (x_1,x_2], \\ |\kappa(x) - C_2| \leqslant \varepsilon, & x \in (x_2,x_3], \\ |\kappa(x) - C_3| \leqslant \varepsilon, & x \in (x_3,x_4], \\ C_i \geqslant 0, & i = 1,2,3 \end{cases}$$

where ε is a small given real number. It is supposed that $P_n(x),\ x \in [x_1,x_4]$ is an n degree polynomial with unknown constant coefficients. Then, if substituting $f(\cdot)$ by $P_n(\cdot)$ in Eq. (14), the sequence of the TNLPP is obtained as follows:

$$\inf I(P_n(\cdot)) = \sum_{i=1}^{4} |P_n(x_i) - y_i|, \tag{15}$$

$$S.t = \begin{cases} |\kappa_n(x) - C_1| \leqslant \varepsilon, & x \in (x_1,x_2], \\ |\kappa_n(x) - C_2| \leqslant \varepsilon, & x \in (x_2,x_3], \\ |\kappa_n(x) - C_3| \leqslant \varepsilon, & x \in (x_3,x_4], \\ C_i \geqslant 0, \quad i = 1,2,3,\ n = 1,2,3,\ldots, \end{cases}$$

where

$$\kappa_n(x) = \frac{|\ddot{P}_n(x)|}{\sqrt{1+\dot{P}_n^2(x)}}.$$

Now, it is supposed that Q is the set of $f(\cdot)$ such that Eq. (16) is feasible and $Q(n)$ is the set of $P_n(\cdot)$ such that Eq. (15) is feasible. Also, it is supposed that Q and $Q(n)$ are not empty. Then, by the following theorem it can be proved that the sequence of solutions for Eq. (15) converges to the solution of Eq. (14).

Theorem 1. *If* $\eta = \inf_Q I(f(\cdot))$ *and* $\eta(n) = \inf_{Q_n} I(P_n(\cdot))$. *Then* $\eta = \lim_{n\to\infty} \eta(n)$.

Proof. Similar to Zamirian et al. [17], it is obvious that $Q(1) \subset Q(2) \subset \cdots \subset Q$, then $\eta(1) \geqslant \eta(2) \geqslant \cdots \geqslant \eta_0$. So, $\{\eta(n)\}$ is a non- increasing and bounded sequence, then it converges to a number called ξ. Set $W = \bigcup_{n=1}^{\infty} Q(n)$; therefore, $\inf I(P_n(\cdot)) = \xi$. Since $W \subset Q$, then $\xi \geqslant \eta$. By the properties of infimum, for every $\varepsilon > 0$, there exists $f(\cdot) \in Q$ such that:

$$\eta < I(f(\cdot)) < \eta + \varepsilon. \tag{16}$$

From the continuity of $I(f(\cdot))$, there is a $\delta > 0$ such that:

$$|I(g(\cdot)) - I(f(\cdot))| < \varepsilon. \tag{17}$$

Whenever for any $g(\cdot) \in Q$, $\|g(\cdot) - f(\cdot)\|_\infty < \delta$ and $(\|g(\cdot) - f(\cdot)\|_\infty = \max_{x\in[x_1,x_4]}(g(x) - f(x)))$. On the other hand, since $f(\cdot) \in C^2(x_1,x_4)$, then there exists the sequence of polynomials as $\{P_n(\cdot)\}$ such that $\{P_n(\cdot)\}$, $\{\dot{P}_n(\cdot)\}$ and $\{\ddot{P}_n(\cdot)\}$ uniformly converge to $\{f_n(\cdot)\}$, $\{\dot{f}_n(\cdot)\}$ and $\{\ddot{f}_n(\cdot)\}$ respectively [18]. Therefore, there is a N belong to N, the set of positive integer numbers, such that for every $n \geqslant N$:

$$\|P_n(\cdot) - f_n(\cdot)\|_\infty < \delta, \tag{18}$$

$$\|\dot{P}_n(\cdot) - \dot{f}_n(\cdot)\|_\infty < \delta, \tag{19}$$

$$\|\ddot{P}_n(\cdot) - \ddot{f}_n(\cdot)\|_\infty < \delta, \tag{20}$$

Now, we claim that there is a $N_1 \geqslant N$ such that for $i = 1, 2, 3, 4$, $|\kappa_{N_1}(x) - C_i| \leqslant \varepsilon;\ \forall x \in (x_i, x_{i+1}]$. Since, otherwise for every $n \geqslant N$, there is a $x \in (x_i, x_{i+1}]$ such that $|\kappa_n(x) - C_i| > \varepsilon$. Thus: $\lim_{n\to\infty}|\kappa_n(x) - C_i| > \varepsilon$ or $|\kappa(x) - C_i| > \varepsilon$, which contradicts the assumption that $f(\cdot) \in Q$.Then:

$$P_N(\cdot) \in Q(N_1) \subset W \subset Q. \tag{21}$$

Using Eqs. (16)–(20), $|I(P_{N_1}(\cdot)) - I(f(\cdot))| < \varepsilon$ or $I(P_{N_1}(\cdot)) < I(f(\cdot)) + \varepsilon < \eta + 2\varepsilon$ or $\eta \leqslant \xi + 2\varepsilon$, so $\xi = \eta$ or $\lim_{n\to\infty}\eta(n) = \eta$. □

Theorem 2. *For every $x \in [x_1, x_4]$, $|\kappa_n(x) - C_i| \leqslant \varepsilon$ if and only if:*

$$\int_{x_1}^{x_4} ||\kappa_n(x) - C_i| - \varepsilon + ||\kappa_n(x) - C_i| - \varepsilon||dx = 0.$$

Proof. Since for every $x \in [x_1, x_4]$, $|\kappa_n(x) - C_i| \leqslant \varepsilon$, then $|\kappa_n(x) - C_i| - \varepsilon \leqslant 0$ or $||\kappa_n(x) - C_i| - \varepsilon| = -|\kappa_n(x) - C_i| - \varepsilon$. So $\int_{x_1}^{x_4} ||\kappa_n(x) - C_i| - \varepsilon + ||\kappa_n(x) - C_i| - \varepsilon||dx = 0$. Then for every $x \in [x_1, x_4]$, $|\kappa_n(x) - C_i| - \varepsilon + ||\kappa_n(x) - C_i| - \varepsilon| = 0$ or $-|\kappa_n(x) - C_i| + \varepsilon = ||\kappa_n(x) - C_i| - \varepsilon|$ so, $|\kappa_n(x) - C_i| - \varepsilon \leqslant 0$ or $|\kappa_n(x) - C_i| \leqslant \varepsilon$.

Therefore, Eq. (15) is equivalent to the following problem:

$$\inf I(P_n(\cdot)),$$

$$S.t. = \begin{cases} \int_{x_1}^{x_2} ||\kappa_n(x) - C_1| - \varepsilon + ||\kappa_n(x) - C_1| - \varepsilon||dx = 0, \\ \int_{x_2}^{x_3} ||\kappa_n(x) - C_2| - \varepsilon + ||\kappa_n(x) - C_2| - \varepsilon||dx = 0, \\ \int_{x_3}^{x_4} ||\kappa_n(x) - C_3| - \varepsilon + ||\kappa_n(x) - C_3| - \varepsilon||dx = 0, \\ C_i \geqslant 0, \quad i = 1, 2, 3,\ n = 1, 2, \ldots \end{cases} \tag{22}$$

For simplicity, we rewrite the above problem as follows:

$$\inf I(P_n(\cdot)),$$

$$S.t. = \begin{cases} \int_{x_1}^{x_2} E_{1n}(x)dx = 0, \\ \int_{x_2}^{x_3} E_{1n}(x)dx = 0, \\ \int_{x_3}^{x_4} E_{3n}(x)dx = 0, \\ C_i \geqslant 0, \quad i = 1, 2, 3,\ n = 1, 2, \ldots, \end{cases} \tag{23}$$

where $E_{in}(x) = ||\kappa_n(x) - C_i| - \varepsilon + ||\kappa_n(x) - C_i| - \varepsilon||$. Now, we partition the intervals $[x_1, x_2]$, $[x_2, x_3]$ and $[x_3, x_4]$ to m_1, m_2 and m_3 parts, respectively, that is $h_1 = \frac{x_2 - x_1}{m_1}$, $h_2 = \frac{x_3 - x_2}{m_2}$ and $h_1 = \frac{x_4 - x_3}{m_3}$. Thus, by using a numerical integration method such as trapezoidal rule, Eq. (23) is converted to the following equation:

$$\min \sum_{i=1}^{4} |P_n(x_i) - y_i|$$

$$S.t. = \begin{cases} \frac{h_1}{2}[E_{1n}(x_1) + E_{1n}(x_1 + h) + \ldots + E_{1n}(x_1 + m_1 h)] = 0, \\ \frac{h_2}{2}[E_{2n}(x_2) + E_{2n}(x_2 + h) + \cdots + E_{2n}(x_2 + m_2 h)] = 0, \\ \frac{h_3}{2}[E_{3n}(x_3) + E_{3n}(x_3 + h) + \cdots + E_{3n}(x_3 + m_3 h)] = 0, \\ C_i \geqslant 0, \quad i = 1, 2, 3,\ n = 1, 2, \ldots \end{cases} \quad \square \tag{24}$$

Theorem 3. The solutions of Eqs. (23) and (24) are the same, if in Eq. (24), m_i $(i = 1, 2, 3)$ tends to infinity, simultaneously.

Proof. See Stor and Bulirsch [19]. □

Eq. (24) is a NLPP with n variables (the unknown constant coefficients of $P_n(\cdot)$) which can be solved using softwares such as Lingo and Matlab.

2.4. TLM method

In this method, the aim is to find a function as $f(\cdot) \in C^2(x_1, x_4)$, such that for every $x \in \lfloor x_j, x_{j+1} \rfloor$, the distance between $f(x)$ and line $L_i(x)$ which connects two given points (x_i, y_i) to (x_{i+1}, y_{i+1}), $i = 1, 2, 3$, is minimized and $\kappa(x)$ is an approximately constant number. These aims are formulated as follows:

$$\inf I(f(\cdot)) = \sum_{i=1}^{4} \int_{x_i}^{x_{i+1}} |f(x) - L_i(x)| dx,$$

$$S.t. \begin{cases} |\kappa(x) - C_1| \leqslant \varepsilon, & x \in [x_1, x_2], \\ |\kappa(x) - C_2| \leqslant \varepsilon, & x \in (x_2, x_3], \\ |\kappa(x) - C_3| \leqslant \varepsilon, & x \in (x_3, x_4], \\ C_i \geqslant 0, \quad i = 1, 2, 3, \ n = 1, 2, \ldots \end{cases} \tag{25}$$

By the same approach with FPM method, $P_n(\cdot)$ is substituted by $f(\cdot)$ in Eq. (25). Then the following equation is achieved:

$$\inf I(P_n(\cdot)) = \sum_{i=1}^{4} \int_{x_i}^{x_{i+1}} |P_n(x) - L_i(x)| dx$$

$$S.t. \begin{cases} |\kappa_n(x) - C_1| \leqslant \varepsilon, & x \in [x_1, x_2], \\ |\kappa_n(x) - C_2| \leqslant \varepsilon, & x \in (x_2, x_3], \\ |\kappa_n(x) - C_3| \leqslant \varepsilon, & x \in (x_3, x_4], \\ C_i \geqslant 0, \quad i = 1, 2, 3, \ n = 1, 2, \ldots \end{cases} \tag{26}$$

Theorem 4. Sequence of the solution of Eq. (26) converges to the solution of Eq. (25).

Proof. The method of proof is the same as Theorem 1. Now, in the same approach for the PM, the following problem is achieved which is equivalent to Eq. (26)

$$\inf \sum_{i=1}^{4} F_{in}(x)$$

$$S.t. \begin{cases} \frac{h_1}{2}[E_{1n}(x_1) + E_{1n}(x_1 + h) + \cdots + E_{1n}(x_1 + m_1 h)] = 0, \\ \frac{h_2}{2}[E_{2n}(x_2) + E_{2n}(x_2 + h) + \cdots + E_{2n}(x_2 + m_2 h)] = 0, \\ \frac{h_3}{2}[E_{3n}(x_3) + E_{3n}(x_3 + h) + \cdots + E_{3n}(x_3 + m_3 h)] = 0 \\ C_i \geqslant 0, \quad i = 1, 2, 3, \ n = 1, 2, \ldots \end{cases} \tag{27}$$

where $F_{in}(x) = \frac{h_i}{2}[F_{in}(x_i) + F_{in}(x_i + h_i) + \cdots + F_{in}(x_i + m_i h_i)]$.

Eq. (27) is a non-linear programming problem which is solved using softwares such as Lingo and Matlab. Now, an algorithm for both FPM and TLM as follows is considered: in this algorithm ε_1 is chosen as a known positive real number which is the error between two consecutive values of target function.

Step 1. Read ε, m_1, m_2, m_3 and set $n = 1$.
Step 2. Solve the NLPP (24).
Step 3. If the NLPP (24) is infeasible then $n = n + 1$ and go to step 2. Else set the value of target function in I_n.
Step 4. Set $n = n + 1$ and solve the NLPP (24) and set the value of goal function I_n.
Step 5. If $|I_n - I_{n-1}| > \varepsilon_1$, then go to step 4.
Step 6. End. □

Considering the schematic interface in VOF method (see Fig. 3(a)), it can be modeled using the presented methods as shown in Fig 3(b) and (c).

To calculate the intersection points coordinates in Fig. 3(c), cells interface lengths are estimated using available formulas [15]. Then, the middle points of two interfaces in adjacent cells (hollow circles in Fig. 3(c)) are used to estimate the free surface curvature. For example, the coordinates of point P for two adjacent cells are estimated as (see Fig. 4)

$$X_p = X(i,j) + 0.5\Delta x - (a_t(i,j) + a_b(i,j+1))/2, \tag{28}$$

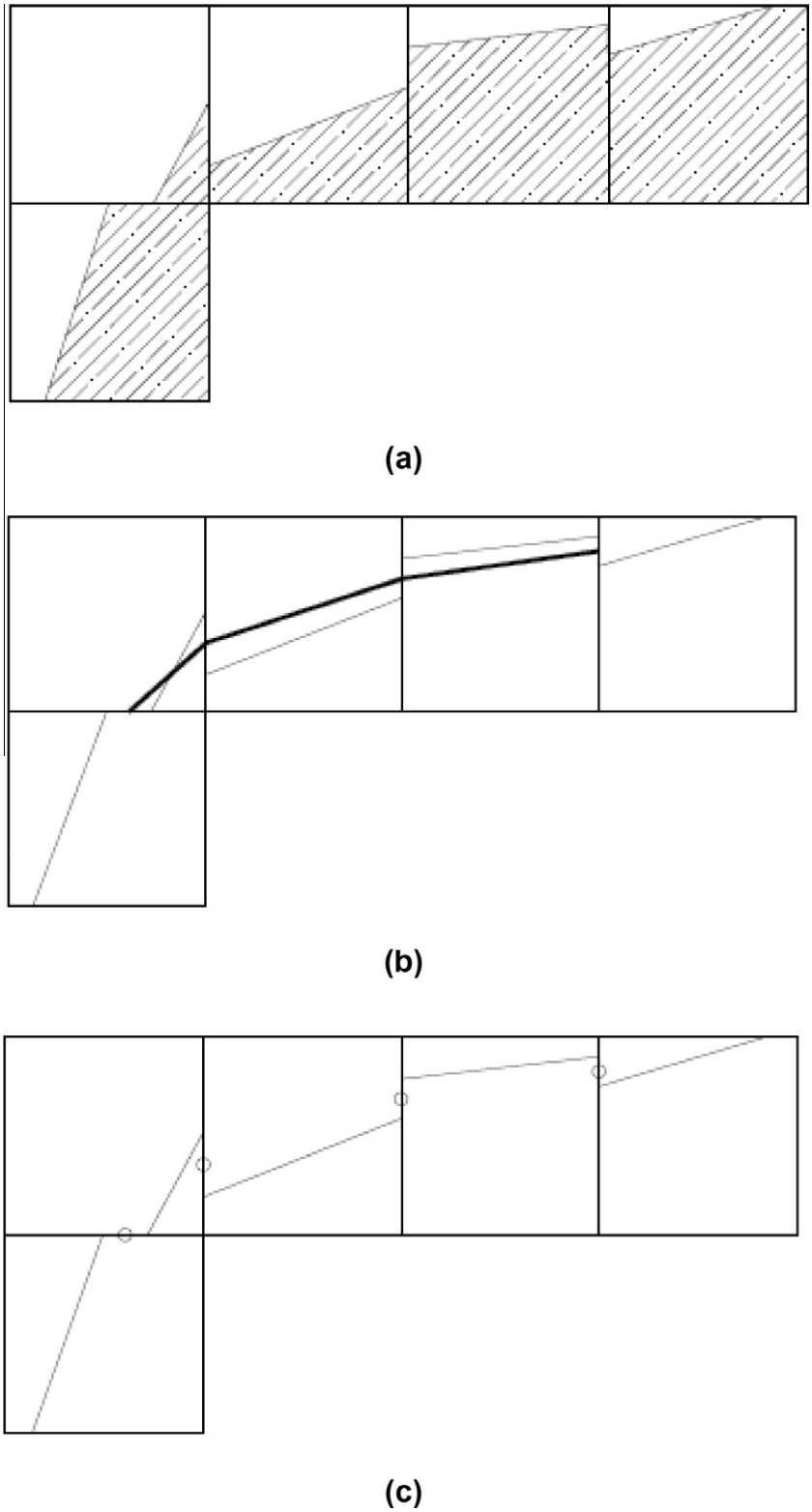

Fig. 3. Schematic interface representation using (a) VOF method; (b) TLM method and (c) FPM method.

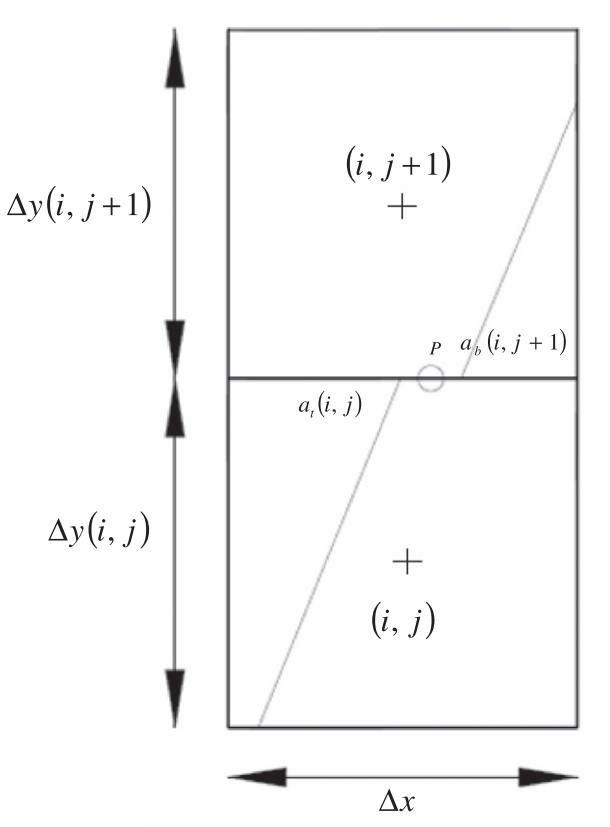

Fig. 4. Coordinates of intersection points for two adjacent cells in FPM and TLM methods.

$$Y_p = Y(i,j) + 0.5\Delta y(i,j), \tag{29}$$

where

$$a_t(i,j) = \left(F(i,j) - \frac{1}{2}\cot an\alpha(i,j)\right)\Delta x, \tag{30}$$

$$a_b(i,j+1) = (2F(i,j+1)\tan\alpha(i,j+1))^{1/2}\Delta x, \tag{31}$$

$$\alpha(i,j) = \tan^{-1}\left[\frac{\Delta x}{\Delta y(i,j)}\tan\beta(i,j)\right], \tag{32}$$

$$\beta(i,j) = \tan^{-1}\left(\frac{-n_x}{n_y}\right), \tag{33}$$

$$n_x = \frac{1}{\Delta x}[F(i+1,j+1) + 2F(i+1,j) + F(i+1,j-1) - F(i-1,j+1) - 2F(i-1,j) - F(i-1,j-1)], \tag{34}$$

$$n_y = \frac{1}{\Delta x}[F(i+1,j+1) + 2F(i,j+1) + F(i-1,j+1) - F(i+1,j-1) - 2F(i,j-1) - F(i-1,j-1)]. \tag{35}$$

Parameter $\alpha(i,j+1)$ is estimated similarly.

3. Numerical results

In this section, four test cases as circle shape, elliptic shape, sinusoidal wave and spurious currents are selected and interface curvature is calculated using presented models.

3.1. Circle shape

As a test problem, a circle with unit radius as shown in Fig. 5 is defined on a uniform mesh.

The curvature of $N = 16$ interface cells (hatched cells in Fig. 5) that are selected randomly is calculated using original and present methods. The effect of mesh size is then evaluated. Therefore, original and present methods are used for calculation of curvature using different mesh sizes as $dx = dy = 0.05$, 0.1, 0.15 and 0.2. The results are compared to analytical ones and shown in Fig. 6. It is clear that in a circle shape, the interface curvature is a constant equivalent to inverse of the radius.

To compare the results, Sum Absolute Error (SAE) and Sum Square Error (SSE) are employed as follows:

$$SAE = \sum_{j=1}^{N} |\kappa_{cal} - \kappa_{exact}|, \tag{36}$$

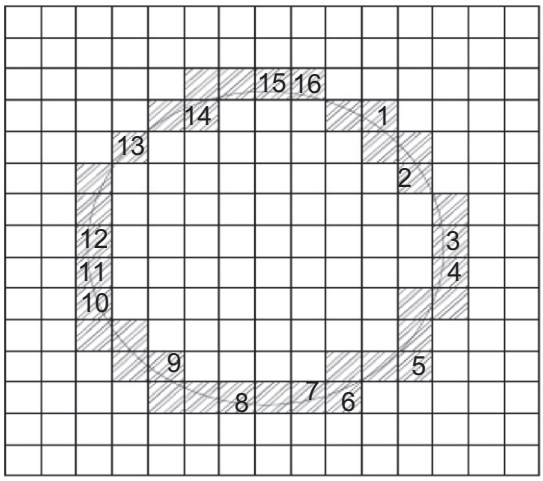

Fig. 5. Schematic circle shape used for assessment of present models.

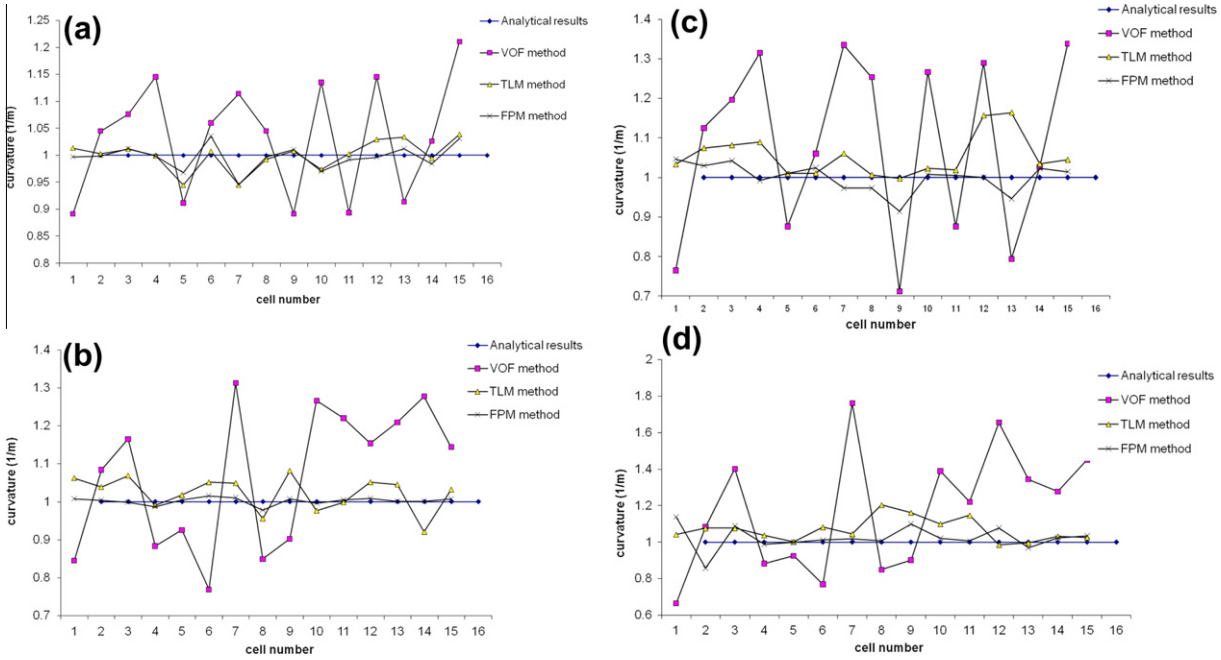

Fig. 6. Comparison of interface cells curvature of circle shape between original and present methods: (a) $dx = dy = 0.05$; (b) $dx = dy = 0.10$; (c) $dx = dy = 0.15$ and (d) $dx = dy = 0.20$.

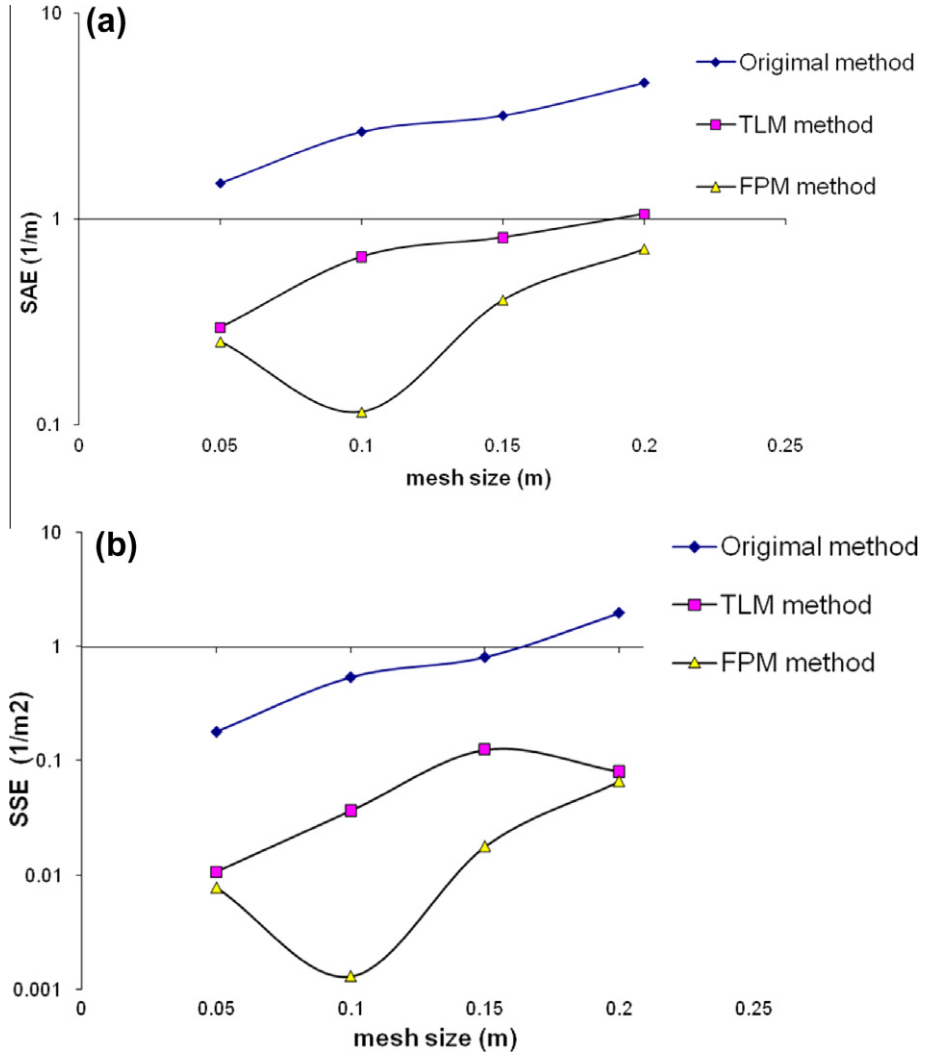

Fig. 7. Comparison of errors of original and present methods for a circle shape: (a) SAE error and (b) SSE error.

$$SSE = \sum_{j=1}^{N} (\kappa_{cal} - \kappa_{exact})^2, \tag{37}$$

κ_{cal} and κ_{exact} are the calculated and analytical curvatures respectively. These errors are calculated for different methods and presented in log scale in Fig. 7.

3.2. Elliptic shape

An elliptic shape as shown in Fig. 8 is defined on a uniform mesh to evaluate the accuracy of present methods. In an elliptic shape, the equation of interface is defined as:

$$\frac{(X - X_0)^2}{a^2} + \frac{(Y - Y_0)^2}{b^2} = 1, \tag{38}$$

where X_0 and Y_0 are the center coordinates and a, b are the elliptic major and minor radius. Here, it is assumed that $X_0 = 0$, $Y_0 = 0$, $a = 0.8$ and $b = 0.6$. The curvature of interface cells is calculated as:

$$\kappa = \frac{|y''|}{(1 + y'^2)^{1.5}}, \tag{39}$$

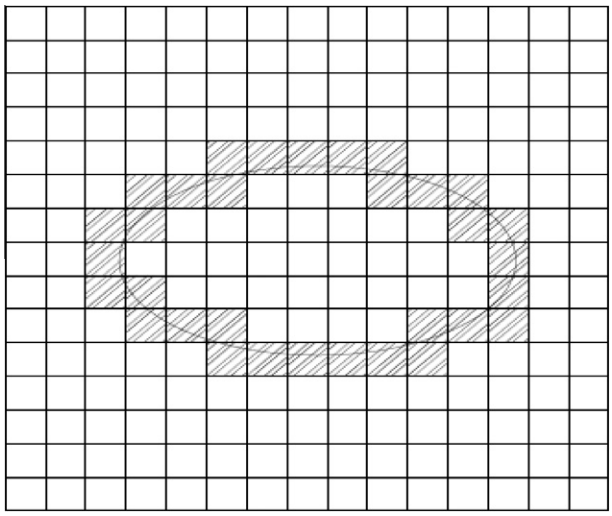

Fig. 8. Schematic elliptic shape in uniform mesh.

where

$$y' = \frac{-b^2}{a^2}(X - X_0)\left[b^2 + Y_0 - \frac{b^2}{a^2}(X - X_0)^2\right]^{-0.5}, \tag{40}$$

$$y'' = \frac{-b^2}{a^2}\left[\left[b^2 + Y_0 - \frac{b^2}{a^2}(X - X_0)^2\right]^{-0.5} + \left(\frac{b}{a}\right)^{-3}(X - X_0)^{-05}\right]. \tag{41}$$

In this section, $N = 12$ interface cells (hatched cells in Fig. 8) are selected randomly and their curvature were calculated using original and present methods. The compared results are shown in Fig. 9.

The aforementioned errors are calculated and presented in log scale in Fig. 10.

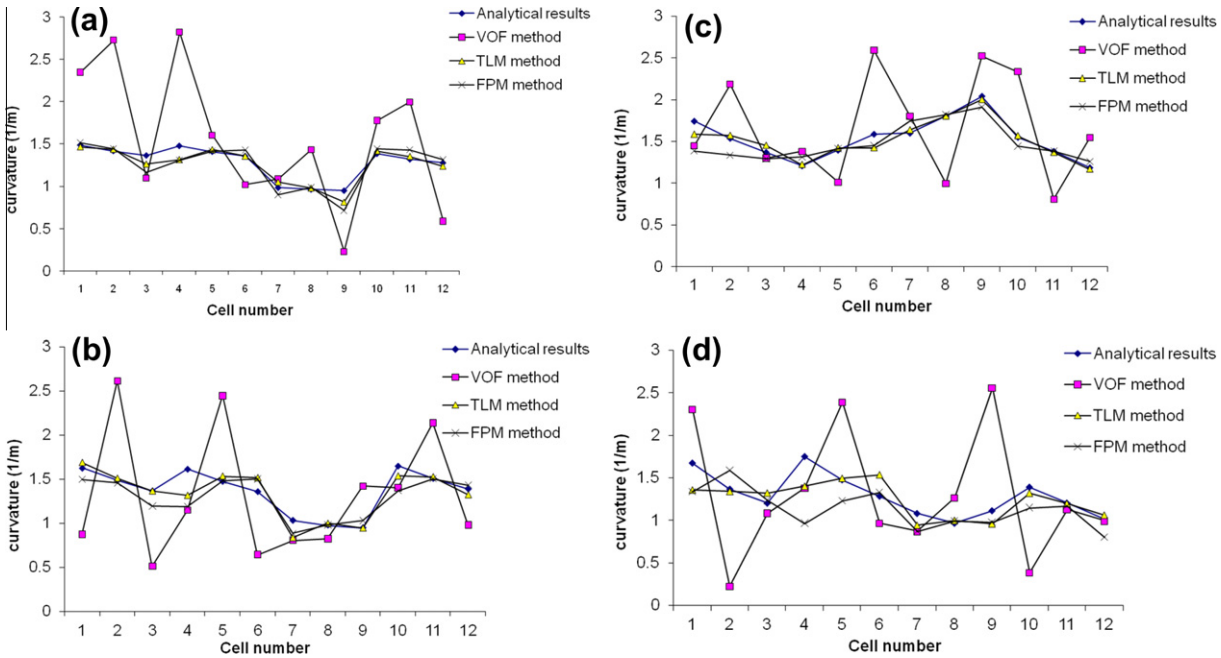

Fig. 9. Comparison of interface cells curvature of elliptic shape between original and present methods: (a) $dx = dy = 0.05$; (b) $dx = dy = 0.10$; (c) $dx = dy = 0.15$ and (d) $dx = dy = 0.20$

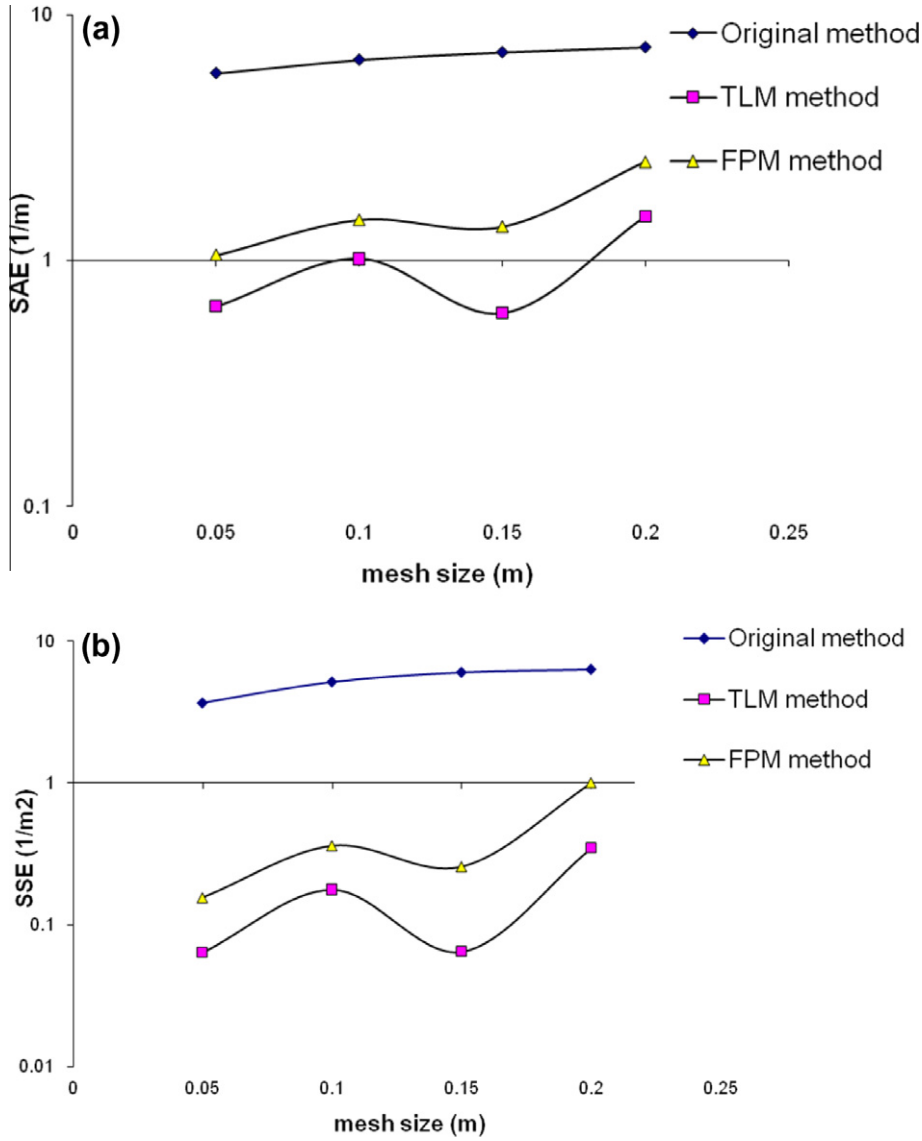

Fig. 10. Comparison of errors of original and present methods for an elliptic shape: (a) SAE error and (b) SSE error.

3.3. Sinusoidal wave

In this section, a sinusoidal wave as a more practical problem is selected for evaluation of the curvature with different methods. Fig. 11 shows the sine wave profile as:

$$\eta = a \sin x \tag{42}$$

in which the amplitude (a) was selected as unity.

In a sinusoidal wave, the curvatures of interface cells are calculated as:

$$\kappa = \frac{|\sin x|}{(1+\cos^2 x)^{1.5}}. \tag{43}$$

To evaluate these curvatures with different methods, $N = 12$ interface cells (hatched cells in Fig. 11) are selected and their curvature were calculated using original and present methods. The mesh size is considered as $dx = dy = 0.1$. The results are presented in Fig. 12.

To compare the accuracy of these methods, the SAE and SSE errors are employed. The results are summarized in Table 1. This table shows that the results of present methods are much better than the original one.

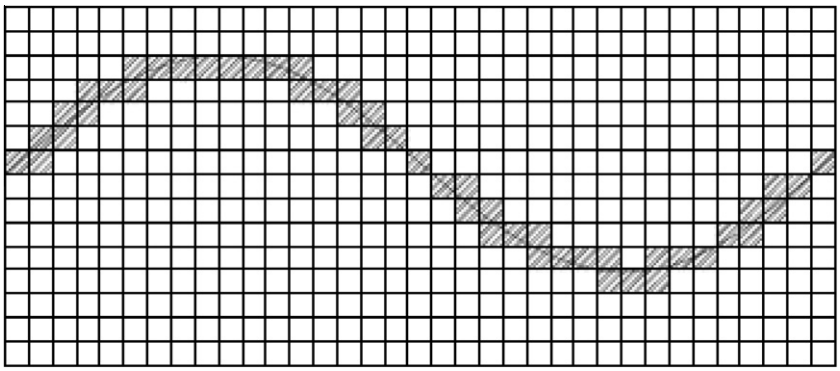

Fig. 11. Schematic sinusoidal wave profile.

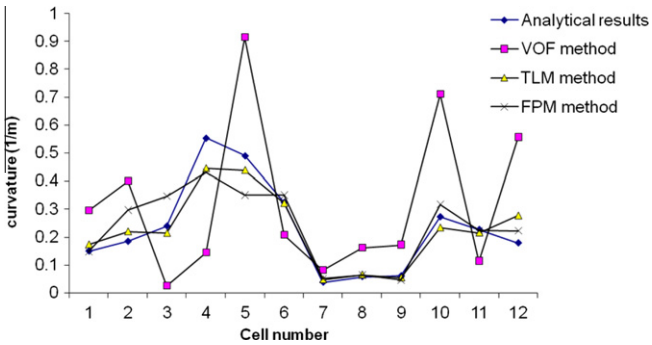

Fig. 12. Comparison of sinosuidal wave curvature for original and present methods.

Table 1
Comparison of errors associated with original and present methods for sinosuidal wave.

Error	VOF	TLM	FPM
SAE	2.7113	0.4198	0.6379
SSE	0.8468	0.0283	0.0641

3.4. Spurious current

Spurious currents are the final test case to show the ability of the present methods for estimation of interface curvature [20,21]. This phenomenon is described as vortices in the neighborhood of interface despite the absence of any external forcing. To model this phenomenon, the fluid is considered to be Newtonian and incompressible. Therefore, 2D continuity and Navier–Stokes equations (NSE) are used as follows:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0, \tag{44}$$

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial x}+(v+v_t)\left[\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right]+S_x, \tag{45}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{1}{\rho}\frac{\partial p}{\partial y}+(v+v_t)\left[\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right]+S_y, \tag{46}$$

where u and v are the velocity components in the x and y directions, ρ is the fluid density, v is the kinematic viscosity, μ is the dynamic viscosity, v_t is the turbulent kinematic viscosity, p is the dynamic pressure and S_i is the source term including acceleration due to gravity in i direction.

In this paper, two-step projection method is used to solve NSE. At the first step, the convective, diffusion and body force terms in the momentum equations are discretized using an explicit scheme. For instance, NSE in the x direction is discretized as follows:

$$(u^{+1}_{i+1/2,j} - u^{n}_{i+1/2,j})/\Delta t = -\left[-\frac{p^{n+1}_{i+1,j} - p^{n+1}_{i,j}}{0.5(\Delta x_i + \Delta x_{i+1})} - Conu_x - Conu_y - Diff_x + S_x\right], \tag{47}$$

where

$$Conu_x = \frac{u^{n}_{i+1/2,j}}{\Delta x_\alpha}[\Delta x_{i+1} Roul + \Delta x_i Rour + \alpha \mathrm{sgn}(u^{n}_{i+1/2,j})(\Delta x_{i+1} Roul - \Delta x_i Rour)], \tag{48}$$

$$Conu_y = \frac{v^{*}_{i+1/2,j}}{\Delta y_\alpha}[\Delta y_{j+1/2} Roub + \Delta y_{j-1/2} Rout + \alpha \,\mathrm{sgn}(u^{n}_{i+1/2,j})(\Delta y_{j+1/2} Roub - \Delta y_{j-1/2} Rout)], \tag{49}$$

$$Diff_x = (v + v_t)^{*}_{i+1/2,j}[4(Rour - Roul)/(\Delta x_i + \Delta x_{i+1}) + 2(Rout - Roub)/(\Delta y_{j-1/2} + \Delta y_{j+1/2}) + (Rovt - Rovb)/\Delta y_j], \tag{50}$$

$$Roul = (u^{n}_{i+1/2,j} - u^{n}_{i-1/2,j})/\Delta x_i, \tag{51}$$

$$Rour = (u^{n}_{i+3/2,j} - u^{n}_{i+1/2,j})/\Delta x_{i+1}, \tag{52}$$

$$Rout = (u^{n}_{i+1/2,j+1} - u^{n}_{i+1/2,j})/\Delta y_{j+1/2}, \tag{53}$$

$$Roub = (u^{n}_{i+1/2,j} - u^{n}_{i+1/2,j-1})/\Delta y_{j-1/2}, \tag{54}$$

$$Rovt = (v^{n}_{i+1,j+1/2} - v^{n}_{i,j+1/2})/\Delta x_{i+1/2}, \tag{55}$$

$$Rovb = (v^{n}_{i+1,j-1/2} - v^{n}_{i,j-1/2})/\Delta x_{i+1/2}, \tag{56}$$

$$v^{*}_{i+1/2,j} = [\Delta x_i (v^{n}_{i+1,j+1/2} + v^{n}_{i+1,j-1/2}) + \Delta x_{i+1}(v^{n}_{i,j+1/2} + v^{n}_{i,j-1/2})]/2(\Delta x_i + \Delta x_{i+1}), \tag{57}$$

$$(v + v_t)^{*}_{i+1/2,j} = [\Delta x_i (v + v_t)_{i+1,j} + \Delta x_{i+1}(v + v_t)_{i,j}]/(\Delta x_i + \Delta x_{i+1}), \tag{58}$$

$$\Delta x_\alpha = \Delta x_i + \Delta x_{i+1} + \alpha \mathrm{sgn}(u^{n}_{i+1/2,j})(\Delta x_{i+1} - \Delta x_i), \tag{59}$$

$$\Delta y_{j+1/2} = 0.5(\Delta y_j + \Delta y_{j+1}), \tag{60}$$

$$\Delta y_{j-1/2} = 0.5(\Delta y_j + \Delta y_{j-1}), \tag{61}$$

$$\Delta x_{i+1/2} = 0.5(\Delta x_i + \Delta x_{i+1}), \tag{62}$$

in which the pressure terms ($p^{n+1}_{i,j}$ and $p^{n+1}_{i+1,j}$) are unknown. Therefore, these terms are omitted from Eq. (47) and intermediate velocity ($\widehat{u}_{i+1/2,j}$) is estimated as follows:

$$\widehat{u}_{i+1/2,j} = u^{n}_{i+1/2,j} - \Delta t\lfloor -Conu_x - Conu_y - Diff_x + S_x \rfloor. \tag{63}$$

In the second step, pressure filed is estimated as:

$$p^{n+1} = p^{n} + \delta p. \tag{64}$$

The Poisson's equation is used to calculate the pressure correction (δp) as:

$$\nabla^2(\delta p) = \frac{\rho}{\Delta t}\nabla \cdot \bar{V}. \tag{65}$$

This equation is discretized using Finite Difference Method (FDM) and solved using Three Diagonal Matrix Algorithm (TDMA). Finally, the velocity field in the new time level is estimated as:

$$u^{n+1}_{i+1/2,j} = u^{n}_{i+1/2,j} - 0.5(\rho(i,j) + \rho(i+1,j))\Delta t \nabla \cdot \delta p. \tag{66}$$

NSE in y direction is discretized and solved in a similar way.

In this procedure, proper selection of Δt can increase the accuracy of results. Therefore time step was selected based on two stability criteria [22,23] as Courant and diffusion conditions respectively as follows:

Table 2
Norms of velocity at 200th time step, $\Delta t = 10^{-5}$ s in spurious current test.

Δx	L_∞	L_2	L_1	Method
1/96	0.00179982	0.00008403	0.00001473	VOF
	0.00154368	0.00008015	0.00001556	TLM
	0.00129615	0.00006416	0.00000945	FPM
1/128	0.00184090	0.00008542	0.00001539	VOF
	0.00156250	0.00008015	0.00001015	TLM
	0.00103342	0.00007386	0.00000776	FPM
1/160	0.00189053	0.00008596	0.00001569	VOF
	0.00109162	0.00007762	0.00001232	TLM
	0.00095231	0.00007081	0.00001199	FPM
1/192	0.00196880	0.00008627	0.00001569	VOF
	0.00109155	0.00007755	0.00001230	TLM
	0.00095194	0.00007078	0.00001225	FPM

$$\Delta t_c = \min\left[\min\left(\frac{\Delta x_i}{|u_{i,j}|}\right), \min\left(\frac{\Delta y_j}{|v_{i,j}|}\right)\right], \tag{67}$$

$$\Delta t_v = \frac{1}{2}\left[\frac{1}{v_e\left[\left(\frac{1}{\Delta x_i}\right)^2 + \left(\frac{1}{\Delta y_j}\right)^2\right]}\right] \tag{68}$$

In the spurious current modeling, computational domain is 1×1 with uniform grid and the time step is $\Delta t = 10^{-5}$ s. The boundary conditions are zero velocity at the top and bottom walls and periodicity in the x and y directions. The initial velocity field is zero. A spherical drop is centered at $(0.5, 0.5)$, with radius $r = 0.125$ and surface tension $\sigma = 0.357$. Both fluids have the same density and viscosity as 4 and 1 respectively. The exact solution is zero velocity for all times. The amplitude Max $|u|$ of the spurious currents is difficult to estimate. But, based on some experimental researches [20,21], they are of the order $0.01\sigma/\mu$, leading to a Reynolds number based on spurious currents, which is of the order $0.01\sigma/Oh^2$. Parameter Oh is the Ohnesorge number estimated as:

$$Oh = (\mu^2/\sigma\rho a)^{1/2}. \tag{69}$$

In this research, CSF method is used to model surface tension. VOF, TLM and FPM methods are then used to estimate interface curvature. To compare the results, norms of the velocity field (L_∞, L_1, L_2) for the spurious currents were used and presented in Table 2 where L_∞ norm gives the maximum speed. L_1 and L_2 norms indicate a measure in an average sense for the computational domain. This table clearly shows the improvement of the results by the present methods relative to the original VOF method.

4. Conclusion and discussion

In this paper, a novel algorithm based on the PM is presented for calculation of curvature on the free surface flows. Two approaches denoted by TLM and FPM are used. The original method was the conventional one used in VOF method. To evaluate these methods, circle, elliptic, sine wave profile and spurious currents are employed. The results show that the present methods demonstrate much better results than the original one. Comparison of the curvature for present methods (FPM and TLM) shows that in situations such as circle shape, the FPM presents better results than TLM. However, in situations such as elliptic shape, TLM demonstrates better estimation. In sum, it can be concluded that the results of curvature estimation for present methods (FPM and TLM) are always more accurate than the results of original VOF method. Comparison of errors in the first three test cases (circle, elliptic and sine wave profile) show that SAE and SSE errors of present methods are decreased by about 76–92% relative to the original VOF ones. Furthermore, the present methods improve the results in spurious current relative to the original VOF method. Therefore, the original method in VOF that estimate curvature based on gradient of colour function can be replaced by the present methods to get more accurate results.

References

[1] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[2] M.M. Francois, S.J. Cummins, E.D. Dendy, D.B. Kothe, J.M. Sicilian, M.W. Williams, A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework, J. Comput. Phys. 213 (1) (2006) 141–173.

[3] D. Jamet, D. Torres, J.U. Brackbill, On the theory and computation of surface tension: the elimination of parasitic currents through energy conservation in the second-gradient method, J. Comput. Phys. 182 (1) (2002) 262–276.
[4] S. Shin, S.I. Abdel-Khalik, V. Daru, D. Juric, Accurate representation of surface tension using the level contour reconstruction method, J. Comput. Phys. 203 (2) (2005) 493–516.
[5] E. Shirani, N. Ashgriz, J. Mostaghimi, Interface pressure calculation based on conservation of momentum for front capturing methods, J. Comput. Phys. 203 (1) (2005) 154–175.
[6] J. Helmsen, P. Colella, E.G. Puckett, Non-convex profile evolution in two dimensions using volume of fluids, Technical Report LBNL-40693, Lawrence Berkeley National Laboratory, 1997.
[7] M. Sussman, A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles, J. Comput. Phys. 187 (1) (2003) 110–136.
[8] S.J. Cummins, M.M. Francois, D.B. Kothe, Estimating curvature from volume fractions, Comput. Struct. 83 (2005) 425–434.
[9] J.Y. Poo, N. Ashgriz, A computational method for determining curvatures, J. Comput. Phys. 84 (2) (1989) 483–491.
[10] Y. Renardy, M. Renardy, PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method, J. Comput. Phys. 183 (2) (2002) 400–421.
[11] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1967) 2182–2189.
[12] C.W. Hirt, B.D. Nichols, Volume of fluid (VoF) methods for dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.
[13] D.L. Youngs, Time dependent multi-material flow with large fluid distortion, in: K.W. Morton, M.J. Baines (Eds.), Numerical Methods for Fluid Dynamics, Academic Press, London, 1982, pp. 237–285.
[14] T. Li, P. Troch, J. De Rouck, A solver for numerical simulation of breaking waves using a cut-cell VOF cell staggered finite-volume approach, Technical Report, Department of Civil Engineering, Ghent University, Belgium, 2003.
[15] M. Rudman, Volume-tracking methods for interfacial flow calculation, Int. J. Numer. Methods Fluids 24 (1997) 671–691.
[16] M. Bussmann, A three dimensional model of an impacting droplet, Ph.D. Thesis, University of Toronto, 2001.
[17] M. Zamirian, A.V. Kamyad, M.H. Farahi, A novel algorithm for solving optimal path planning problems based on parametrization method and fuzzy aggregation, Phys. Lett. A 373 (2009) 3439–3449.
[18] W. Rudin, Principles of Mathematical Analysis, third ed., McGraw-Hill, 1976.
[19] J. Store, R. Bulirsch, Introduction to Numerical Analysis, Springer-Verlag, New York, 1992.
[20] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, G. Zanetti, Modelling merging and fragmentation in multiphase flows with SURFER, J. Comput. Phys 113 (1994) 134–147.
[21] R. Scardovelli, S. Zaleski, Direct numerical simulation of free surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.
[22] U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411.
[23] D. Geuyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, J. Comput. Phys. 152 (1999) 423–456.